

# Design of fuzzy rule based classifier using the monkey algorithm<sup>1</sup>

**Ilya A. Hodashinsky**

*Professor, Department of Complex Information Security of Computer Systems  
Tomsk State University of Control Systems and Radioelectronics (TUSUR)  
Address: 40, Prospect Lenina, Tomsk, 634050, Russian Federation  
E-mail: hodashn@rambler.ru*

**Sergey S. Samsonov**

*Student, Department of Complex Information Security of Computer Systems  
Tomsk State University of Control Systems and Radioelectronics (TUSUR)  
Address: 40, Prospect Lenina, Tomsk, 634050, Russian Federation  
E-mail: samsonicx@mail.ru*

## Abstract

This article presents an approach for building fuzzy rule based classifiers. A fuzzy rule-based classifier consists of IF-THEN rules with fuzzy antecedents (IF-part) and the class marks in consequents (THEN-part). Antecedent parts of the rules break down the input feature space into a set of fuzzy areas, and consequents define the classifier exit, marking these areas with a class mark. Two main phases of building the classifier are selected: generating the fuzzy rule base and optimizing the rule antecedent parameters. The classifier structure was formed by an algorithm for generating the rule base by extreme features found in the training sample. The peculiarity of this algorithm is that it is generated according to one classification rule for each class. The rule base formed by this algorithm has as low as practicable size in classification of a given data set. The optimization of parameters of antecedents of the fuzzy rules is implemented using the monkey algorithm adapted for these purposes, which is based on observations of monkey migration in the highlands. In the process of the algorithm work, three operations are performed: climb process, watch jump process and somersault process. One of the algorithm's advantages in solution of high-dimension optimization problems is calculation of the pseudo-gradient of the objective function. Irrespective of the dimension at each iteration of the algorithm execution only two values of the objective function are to be calculated.

The effectiveness of fuzzy rule-based classifiers built with the use of the proposed algorithms was checked on actual data from the KEEL-dataset repository. A comparative analysis was conducted using the known analog algorithms "D-MOFARC" and "FARC-HD". The number of rules used by the classifiers built with the use of the algorithms so developed is much lower than the number of rules in analog classifiers with a comparable classification accuracy, that points to the highest interpretability of the classifiers built with the use of the proposed approach.

**Key words:** fuzzy classifier, optimization of fuzzy parameters, monkey algorithm, fuzzy rule extraction.

**Citation:** Hodashinsky I.A., Samsonov S.S. (2017) Design of fuzzy rule based classifier using the monkey algorithm. *Business Informatics*, no. 1 (39), pp. 61–67. DOI: 10.17323/1998-0663.2017.1.61.67.

## Introduction

Fuzzy rule based classifiers belong to a class of fuzzy rule-based systems. Classifiers of this type are widely used in modern business applications

due to their ability to manage uncertainty, inaccuracy and incompleteness of information [1], for example, in such areas as credit risk assessment [2, 3], marketing [4, 5], electronic business and e-commerce [6]. The

<sup>1</sup> This work was supported by the Ministry of Education and Science of the Russian Federation, agreement no. 8.9628.2017/BP

advantages of the fuzzy rule-based classifiers include their good interpretability [7] and lack of assumptions required for statistical classification [8].

Construction of fuzzy rule-based classifiers involves the solution of two main problems: generating the fuzzy rule base and optimizing the rule antecedent parameters (IF-parts). To generate a fuzzy rule base, clustering algorithms are most often used, resulting in formation of the initial “rude” approximation of the fuzzy rule-based classifier. The procedure for optimization of rule antecedent parameters or “fine” tuning is generally performed through derivatives, swarm intelligence algorithms or evolutionary computations [1, 2, 9–14]. For solution of the problems listed above, this paper proposes to use the algorithm of generating the rule base by extreme features and the monkey algorithm.

Application of the algorithm for generating the rule base for extreme values of features enables us to minimize the number of rules to the number of classes, and so to increase the interpretability of the result.

The monkey algorithm is based on the observation of monkey migration in the highlands [15]. In the process of the algorithm work, three operations are performed: climb process, watch jump process and somersault process. One of the algorithm advantages in solution of high-dimension optimization problems is calculation of the pseudo-gradient of the objective function. Irrespective of the dimension at each iteration of the algorithm execution, only two values of the objective function are to be calculated [16].

The purpose of this paper is to describe the algorithms of building fuzzy rule-based classifiers: the algorithm for generating rule base by extreme features and the monkey algorithm. Application of these algorithms is aimed at increasing the accuracy in solving classification problems, while maintaining the interpretability of the solution obtained.

**1. Statement of the problem**

Let us assume that we have universum  $U = (A, C)$ , where  $A = \{x_1, x_2, \dots, x_n\}$  is a set of input features,  $C = \{c_1, c_2, \dots, c_m\}$  is a set of classes. Suppose  $\mathbf{X} = x_1 \cdot x_2 \cdot \dots \cdot x_n \in \mathfrak{R}^n$  is an n-dimensional space of feature values. Object  $u$  in this universum is characterized by its vector of feature values. The classification problem consists in prediction of object class  $u$  by its vector of feature values.

The traditional classifier can be defined as a function of

$$f : \mathfrak{R}^n \rightarrow \{0,1\}^m,$$

where  $f(\mathbf{x};\boldsymbol{\theta}) = (c_1, c_2, \dots, c_m)$ , where  $c_i = 1$ , and  $c_j = 0$  ( $j \in [1, m], i \neq j$ ), when the object specified by vector  $\mathbf{x}$  belongs to class  $c_j$ ;

$\boldsymbol{\theta}$  – vector of the classifier parameters.

The fuzzy rule-based classifier can be presented in a functional form, which assigns a point in the input feature space with a class mark with the calculated level of confidence:

$$f : \mathfrak{R}^n \rightarrow [0,1]^m.$$

The fuzzy rule-based classifier is based on a production rule of the form:

$$R_{ij} : \text{IF } x_1=A_{1i} \text{ И } x_2=A_{2i} \text{ И } x_3=A_{3i} \text{ И } \dots \text{ И } x_n=A_{ni} \\ \text{THEN class}=c_j,$$

where  $A_{ki}$  – fuzzy term characterizing the  $k$ -th feature in the  $i$ -th rule ( $i \in [1, R]$ );

$R$  – number of rules.

In this paper, the class is determined on the principle “a winning team gets everything”:

$$\text{class} = c_{j^*}, j^* = \arg \max_{1 \leq j \leq m} \beta_j,$$

where  $\beta_j(\mathbf{x}) = \sum_{R_{ij}} \prod_{k=1}^n \mu_{A_{jk}}(x_k)$ ,  $j = 1, 2, \dots, m$ ;

$\mu_A(\cdot)$  – membership function of fuzzy term  $A$ .

Let us assume that there is a table of observations  $\{(\mathbf{x}_p; c_p), p = 1, \dots, Z\}$ . Let us determine the following unit function:

$$\text{delta}(p, \boldsymbol{\theta}) = \begin{cases} 1, & \text{if } c_p = f(c_p, \boldsymbol{\theta}) \\ 0, & \text{otherwise} \end{cases}, p = 1, 2, \dots, Z.$$

Then the fitness function or the classifier precision measure can be expressed in the following way:

$$E(\boldsymbol{\theta}) = \frac{\sum_{p=1}^Z \text{delta}(p, \boldsymbol{\theta})}{Z}.$$

The problem of building a fuzzy rule-based classifier comes down to finding the maximum of the specified function in space  $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_D)$ :

$$\max(E(\boldsymbol{\theta})), \theta_i \in \{ \theta_i : \theta_{i,\min} < \theta_i < \theta_{i,\max}, i = 1, 2, \dots, D \},$$

where  $\theta_i$  – value of parameter  $\theta_i$  from the interval  $[\theta_{i,\min}, \theta_{i,\max}]$ ;

$\theta_{i,\min}, \theta_{i,\max}$  – lower and upper limits of each parameter, respectively.

Figure 1 depicts an example explaining the formation of vector  $\boldsymbol{\theta}$ . At this time, variable  $x_1$  is presented by three

triangular terms, each of which is represented by three parameters  $(a, b, c)$ , included in vector  $\theta = (a_{11}, b_{11}, c_{11}, a_{12}, b_{12}, c_{12}, a_{13}, b_{13}, c_{13}, \dots, a_{21}, b_{21}, c_{21}, \dots)$ .

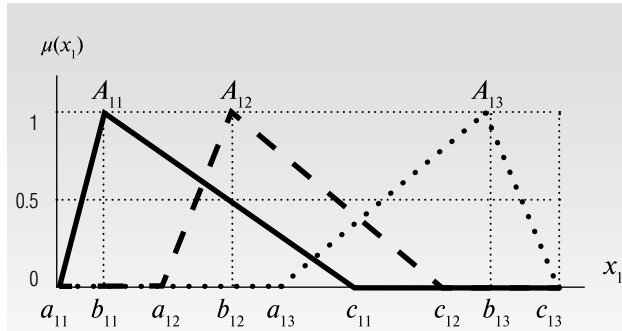


Fig. 1. Fuzzy partition of variable  $x_1$

For finding optimal parameters  $\theta$ , we propose using the monkey algorithm.

### 2. Monkey algorithm

The monkey algorithm (MA) is a metaheuristic optimization algorithm simulating the migration of a monkey population in the highlands. Seven main stages of algorithms are considered below.

#### 1) Solutions coding

First,  $M$  – a magnitude of monkey population is determined, in which the position of each  $i$ -th monkey presents the solution specified by vector  $\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{iD})$ ,  $i = 1, \dots, M$ .

#### 2) Initialization of population

Possible positions of monkeys in a  $D$ -dimensional hypercube are generated in random manner, or position-solutions are given by the user. A mixed strategy of initialization is also possible, when one part of the population is defined by the user, and the other part is randomly generated.

#### 3) Climb process

I. For each  $i$ -th monkey vector  $\Delta\theta_i = (\Delta\theta_{i1}, \Delta\theta_{i2}, \dots, \Delta\theta_{iD})$  is generated, where

$$\Delta\theta_{ij} = \begin{cases} a, & \text{if } \text{rand}(0;1) \geq 0,5 \\ -a, & \text{if } \text{rand}(0;1) < 0,5 \end{cases}, i = 1, \dots, M, j = 1, \dots, D,$$

$a > 0$  is a step length.

II. Calculate

$$E'_{ij}(\theta_i) = \frac{E(\theta_i + \Delta\theta_i) - E(\theta_i - \Delta\theta_i)}{2\Delta\theta_{ij}}, i = 1, \dots, M, j = 1, \dots, D.$$

Vector  $E'_i(\theta_i) = (E'_{i1}(\theta_i), E'_{i2}(\theta_i), \dots, E'_{iD}(\theta_i))$  is a pseudo-gradient of fitness function  $E(\cdot)$  in point  $\theta_i$ .

III. Calculate  $z_j = \theta_{ij} + a \cdot \text{sign}(E'_{ij}(\theta_i))$  and form vector  $z = (z_1, z_2, \dots, z_D)$ .

IV. If the obtained solution vector  $z$  is compatible with the limits of building the fuzzy rule-based classifier, vector  $\theta_i$  is replaced with vector  $z$ ; otherwise, vector  $\theta_i$  remains unchanged.

V. Repeat Steps I–IV a specified number of times.

#### 4) Watch jump process

I. Form vector  $z = (z_1, z_2, \dots, z_D)$  of the randomly generated uniformly distributed real numbers in the range  $(\theta_{ij} - b, \theta_{ij} + b)$ , where  $b$  is a parameter characterizing the monkey's ability to observe.

II. If value  $E(z) > E(\theta_i)$  and vector  $z$  are compatible with the requirements of constructing a fuzzy rule-based classifier, then vector  $\theta_i$  is replaced with vector  $z$ .

III. Repeat Steps I–II a specified number of times.

#### 5) Somersault process

I. Generate a random evenly distributed real number  $\alpha$  from the interval  $[c, d]$ , where  $c, d$  are algorithm parameters.

II. Calculate  $z_j = \theta_{ij} + \alpha(\rho_j - \theta_{ij})$ ,  $j = 1, 2, \dots, D$ ,

$$\text{where } \rho_j = \frac{\sum_{i=1}^M \theta_{ij}}{M}, i = 1, \dots, M, j = 1, \dots, D.$$

III. If the obtained vector  $z = (z_1, z_2, \dots, z_D)$  is compatible with the requirements of constructing a fuzzy rule-based classifier, and value  $E(z) > E(\theta_i)$ , then vector  $\theta_i$  is replaced with vector  $z$ ; otherwise, vector  $\theta_i$  remains unchanged.

IV. Repeat Steps I–III a specified number of times.

#### 6) Repeat $N$ times the climb process operators, watch jump and somersault processes

#### 7) Output of the best solution

### 3. Algorithm for generating rule base by extreme features

The algorithm for generating rule base by extreme features (EC) is intended to form the initial rule base of the fuzzy rule-based classifier containing one rule at a time for each class. The rules are laid down based on extreme values of training sample  $\{(x_p; t_p), p = 1, \dots, Z\}$ . Let us introduce the following designations:  $m$  is a number of classes,  $n$  is a number of features,  $\Omega^*$  is a classifier rule base.

Table 1.

**Description of data sets**

| Group | Name            | Features | Copies | Classes |
|-------|-----------------|----------|--------|---------|
| SS    | 1. haberman     | 3        | 306    | 2       |
|       | 2. iris         | 4        | 150    | 3       |
|       | 3. balance      | 4        | 625    | 3       |
|       | 4. newthyroid   | 5        | 215    | 3       |
|       | 5. bupa         | 6        | 345    | 2       |
|       | 6. pima         | 8        | 768    | 2       |
|       | 7. glass        | 9        | 214    | 7       |
|       | 8. Wisconsin    | 9        | 683    | 2       |
| SL    | 9. banana       | 2        | 5300   | 2       |
|       | 10. titanic     | 3        | 2201   | 2       |
|       | 11. phoneme     | 5        | 5404   | 2       |
|       | 12. magic       | 10       | 19020  | 2       |
|       | 13. page-blocks | 10       | 5472   | 5       |
| LS    | 14. wine        | 13       | 178    | 3       |
|       | 15. Cleveland   | 13       | 297    | 5       |
|       | 16. heart       | 13       | 270    | 2       |
|       | 17. hepatitis   | 19       | 80     | 2       |
| LL    | 18. segment     | 19       | 2310   | 7       |
|       | 19. twonorm     | 20       | 7400   | 2       |
|       | 20. thyroid     | 21       | 7200   | 3       |

Input:  $m, \{(x_p; t_p)\}$ .

Output: Classifier rule base  $\Omega^*$ .

$\Omega := \emptyset$ ;

Loop on  $j$  from 1 to  $m$

Loop on  $k$  from 1 to  $n$

search min  $class_{jk} := \min(x_{pk})$ ;

search max  $class_{jk} := \max(x_{pk})$ ;

forming fuzzy term  $A_{jk}$ , covering the interval

$[\min class_{jk}, \max class_{jk}]$ ;

Loop end

Creating rule  $R_{ij}$  based on terms  $A_{jk}$ , referring

the observation to a class with identifier  $c_j$ ;

$\theta^* := \theta \cup \{R_{ij}\}$ ;

Loop end

Output  $\theta^*$ .

Table 2.

Comparison of classifiers on KEEL data sets

| Data set |                 | EC+MA |      |             | D-MOFARC |       |             | FARC-HD |       |             |
|----------|-----------------|-------|------|-------------|----------|-------|-------------|---------|-------|-------------|
| Group    | Name            | #R    | #L   | #T          | #R       | #L    | #T          | #R      | #L    | #T          |
| SS       | 1. haberman     | 2     | 79.2 | <b>73.8</b> | 9.2      | 81.7  | 69.4        | 5.7     | 79.2  | 73.5        |
|          | 2. iris         | 3     | 97.8 | 95.3        | 5.6      | 98.1  | <b>96.0</b> | 4.4     | 98.6  | 95.3        |
|          | 3. balance      | 3     | 87.5 | 86.7        | 20.1     | 89.4  | 85.6        | 18.8    | 92.2  | <b>91.2</b> |
|          | 4. newthyroid   | 3     | 97.5 | 90.7        | 9.5      | 99.8  | <b>95.5</b> | 9.6     | 99.2  | 94.4        |
|          | 5. bupa         | 2     | 74.2 | 68.4        | 7.7      | 82.8  | <b>70.1</b> | 10.6    | 78.2  | 66.4        |
|          | 6. pima         | 2     | 75.5 | 71.3        | 10.4     | 82.3  | 75.5        | 20.2    | 82.3  | 76.2        |
|          | 7. glass        | 7     | 69.0 | 61.3        | 27.4     | 95.2  | <b>70.6</b> | 18.2    | 79.0  | 69.0        |
|          | 8. wisconsin    | 2     | 96.8 | 96.4        | 9.0      | 98.6  | <b>96.8</b> | 13.6    | 98.3  | 96.2        |
| SL       | 9. banana       | 2     | 78.9 | 78.4        | 8.7      | 90.3  | <b>89.0</b> | 12.9    | 86.0  | 85.5        |
|          | 10. titanic     | 2     | 78.5 | 78.0        | 10.4     | 78.9  | 78.7        | 4.1     | 79.1  | <b>78.8</b> |
|          | 11. phoneme     | 2     | 79.9 | 79.3        | 9.3      | 84.8  | <b>83.5</b> | 17.2    | 83.9  | 82.4        |
|          | 12. magic       | 2     | 81.2 | 81.0        | 32.2     | 86.3  | <b>85.4</b> | 43.8    | 85.4  | 84.8        |
|          | 13. page-blocks | 5     | 95.5 | 95.3        | 21.5     | 97.8  | <b>97.0</b> | 18.4    | 95.5  | 95.0        |
| LS       | 14. wine        | 3     | 99.0 | <b>96.6</b> | 8.6      | 100.0 | 95.8        | 8.3     | 100.0 | 95.5        |
|          | 15. cleveland   | 5     | 60.7 | 57.1        | 45.6     | 90.9  | 52.9        | 42.1    | 82.2  | <b>58.3</b> |
|          | 16. heart       | 2     | 75.8 | 74.4        | 18.7     | 94.4  | <b>84.4</b> | 27.8    | 93.1  | 83.7        |
|          | 17. hepatitis   | 2     | 94.1 | 77.8        | 11.4     | 100.0 | <b>90.0</b> | 10.4    | 99.4  | 88.7        |
| LL       | 18. segment     | 7     | 85.8 | 84.0        | 26.2     | 98.0  | <b>96.6</b> | 41.1    | 94.8  | 93.3        |
|          | 19. twonorm     | 2     | 97.5 | <b>97.1</b> | 10.2     | 94.5  | 93.1        | 60.4    | 96.6  | 95.1        |
|          | 20. thyroid     | 3     | 99.6 | <b>99.3</b> | 5.9      | 99.3  | 99.1        | 4.9     | 94.3  | 94.1        |

#### 4. Experiment

For assessing the operating efficiency of the fuzzy rule-based classifiers optimized by a combination of the above algorithms (EC+MA), tests were conducted on data sets from the KEEL repository given in *Table 2*. Each data set was selected according to one of the following groups:

- ◆ “small number of features – small number of copies” (SS): data sets with a number of features less than 13 and a number of copies less than 1000;
- ◆ “small number of features – large number of copies” (SL): data sets with a number of features less than 13 and a number of copies more than or equal to 1000;
- ◆ “large number of features – small number of copies” (LS): data sets with a number of features more than or equal to 13 and a number of copies less than 1000;
- ◆ “large number of features – large number of copies” (LL): data sets with a number of features more than or equal to 13 and a number of copies more than or equal to 1000.

The experiments have been implemented in accordance with the principle of cross-validation, which intends to separate the data set into training and test data. As per the described principle, each data set is represented by a group of files being test and training samples. Accordingly, during the experiments the classifier was constructed on training samples, following which the accuracy was evaluated on test samples. The total value of accuracy on the test and training data was determined by calculation of the average value.

For all data sets, triangular membership functions were used. As the algorithm parameters, the following parameters were selected: the number of species is 30; climb process iterations are 5; jump iterations are 5; roll-over iterations are 15, the watch-jump process interval is 0.5, boundaries of the somersault process are 0.5 and 0.5 for the left and right boundaries, respectively.

*Table 2* shows the average results of the experimental study of the monkey algorithm when constructing fuzzy rule-based classifiers on a full feature set, as well as the results of analog algorithms “D-MOFARC” and “FARC-HD” [11]; where #R is the number of rules, #L is the percentage of correct classification on the training sample, #T is the percentage of correct classification on the test sample; the best results are in semi-bold type.

To assess the statistical significance of differences in accuracy and the number of rules of classifiers formed by a combination of algorithms EC+MA and analog classifiers, a criterion of pairwise comparisons Wilcoxon–Mann–Whitney was used.

Comparative analysis made it possible to draw the following conclusions:

- 1) The Wilcoxon–Mann–Whitney test indicates a significant difference between the number of rules in the classifiers based on EC+MA and analog classifiers ( $p\text{-value} < 5E-8$ );
- 2) The Wilcoxon–Mann–Whitney test indicates the absence of a significant difference between the accuracy of classification in the compared classifiers.

These findings lead us to the following conclusion: with statistically undistinguished accuracy of the compared classifiers, the classifiers optimized by a combination of algorithms EC+MA are preferable due to the smaller number of rules, and that ultimately points to their possible higher interpretability.

An important issue of algorithm comparison based on their computational complexity remained beyond the framework of article, because the articles which provide the results of previous studies have no detailed description of the algorithms and do not present the experiment results by which we can judge the computational complexity of these approaches.

#### Conclusion

This paper addresses methods of building fuzzy rule-based classifiers. The classifier structure was formed by an algorithm for generating rule base by extreme features. The monkey algorithm was applied to optimize the classifier parameters.

The efficiency of the fuzzy rule-based classifiers configured by the listed algorithms was checked on a number of data sets from the KEEL repository. The classifications obtained have a good learning capability (a high percentage of correct classification on training samples) and as much as good predictive capability (a high percentage of correct classification on test samples).

The number of rules used by the classifiers built with the use of the developed algorithms is much less than the number of rules in the analog classifiers with a comparable classification accuracy; that points to a possibly higher interpretability of classifiers built on the basis of combination EC+MA. ■

## References

1. Garcia-Galan S., Prado R.P., Exposito M.J.E. (2015) Rules discovery in fuzzy classifier systems with PSO for scheduling in grid computational infrastructures. *Applied Soft Computing*, no. 29, pp. 424–435.
2. Gorzalczany M.B., Rudzinski F. (2016) A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability. *Applied Soft Computing*, no. 40, pp. 206–220.
3. Laha A. (2007) Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. *Advanced Engineering Informatics*, no. 21, pp. 281–291.
4. Zhao R., Chai C., Zhou X. (2012) Using evolving fuzzy classifiers to classify consumers with different model architectures. *Physics Procedia*, no. 25, pp. 1627–1636.
5. Setnes M., Kaymak U. (2001) Fuzzy modeling of client preference from large data sets: An application to target selection in direct marketing. *IEEE Transactions on Fuzzy Systems*, no. 9, pp. 153–163.
6. Meier A., Werro N. (2007) A fuzzy classification model for online customers. *Informatica*, no. 31, pp. 175–182.
7. Gorbunov I.V., Hodashinsky I.A. (2015) Metody postroeniya trekhkriterial'nykh pareto-optimal'nykh nechetkikh klassifikatorov [Methods of buildings Pareto-optimal fuzzy classifiers]. *Artificial Intelligence and Decision Making*, no. 2, pp. 75–87 (in Russian).
8. Scherer R. (2012) Multiple fuzzy classification systems. *Studies in Fuzziness and Soft Computing*, vol. 288. Berlin: Springer-Verlag.
9. Hodashinsky I.A. (2012) Identifikatsiya nechetkikh sistem na baze algoritma imitatsii otzhiga i metodov, osnovannykh na proizvodnykh [Simulated annealing and methods based on derivatives for fuzzy system identification]. *Information Technologies*, no. 3, pp. 14–20 (in Russian).
10. Antonelli M., Ducange P., Marcelloni F. (2014) An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets. *Neurocomputing*, no. 146, pp. 125–136.
11. Fazzolari F., Alcalá R., Herrera F. (2014) A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm. *Applied Soft Computing*, no. 24, pp. 470–481.
12. Hodashinsky I.A., Gorbunov I.V. (2012) Optimizatsiya parametrov nechetkikh sistem na osnove modifitsirovannogo algoritma pchelinykh kolonii [Optimization of fuzzy systems parameters using the modified bee colonies algorithm]. *Mechatronics, Automation, Control*, no. 10, pp. 15–20 (in Russian).
13. Hodashinsky I.A., Dudin P.A. (2011) Identifikatsiya nechetkikh sistem na osnove pryamogo algoritma murav'inykh kolonii [Fuzzy systems identification based on direct ant colony algorithm]. *Artificial Intelligence and Decision Making*, no. 3, pp. 26–33 (in Russian).
14. Hodashinsky I.A., Dudin P.A. (2008) Parametricheskaya identifikatsiya nechetkikh modeley na osnove gibridnogo algoritma murav'inykh kolonii [Parametric fuzzy model identification based on a hybrid ant colony algorithm]. *Aviometriya*, no. 5 (44), pp. 24–35 (in Russian).
15. Zhao R., Tang W. (2008) Monkey algorithm for global numerical optimization. *Journal of Uncertain Systems*, no. 2, pp. 165–176.
16. Zheng L. (2013) An improved monkey algorithm with dynamic adaptation. *Applied Mathematics and Computation*, no. 222, pp. 645–657.

## Построение нечеткого классификатора на основе алгоритма обезьян<sup>2</sup>

### И.А. Ходашинский

доктор технических наук

профессор кафедры комплексной информационной безопасности электронно-вычислительных систем  
Томский государственный университет систем управления и радиоэлектроники (ТУСУР)

Адрес: 634050, г. Томск, пр-т Ленина, д. 40

E-mail: hodashn@rambler.ru

### С.С. Самсонов

студент кафедры комплексной информационной безопасности электронно-вычислительных систем

Томский государственный университет систем управления и радиоэлектроники (ТУСУР)

Адрес: 634050, г. Томск, пр-т Ленина, д. 40

E-mail: samsonicx@mail.ru

<sup>2</sup> Исследование выполнено в рамках базовой части государственного задания министерства образования и науки Российской Федерации на 2017-2019 гг. № 8.9628.2017/БЧ

**Аннотация**

В статье представлен подход к построению классификаторов на основе нечетких правил. Нечеткий классификатор состоит из ЕСЛИ-ТО правил с нечеткими антецедентами (ЕСЛИ-часть) и метками класса в консеквентах (ТО-часть). Антецедентные части правил разбивают входное пространство признаков на множество нечетких областей, а консеквенты задают выход классификатора, помечая эти области меткой класса. Выделены два основных этапа построения классификатора: генерация базы нечетких правил и оптимизация параметров антецедентов правил. Формирование структуры классификатора выполнялась алгоритмом генерации базы правил по экстремальным значениям признаков, найденным в обучающей выборке. Особенность данного алгоритма заключается в том, что он генерирует по одному классифицирующему правилу на каждый класс. База правил, сформированная данным алгоритмом, имеет минимально возможный размер при классификации заданного набора данных. Оптимизация параметров антецедентов нечетких правил выполнена с помощью адаптированного для этих целей алгоритма обезьян, основанного на наблюдениях за передвижением обезьян в горной местности. В процессе работы алгоритма выполняются три оператора: движение вверх, локальный прыжок и глобальный прыжок. Одним из достоинств алгоритма при решении задач оптимизации большой размерности является вычисление псевдо-градиента целевой функции, причем вне зависимости от размерности на каждой итерации выполнения алгоритма требуется вычислить только два значения целевой функции.

Эффективность нечетких классификаторов, построенных с помощью предложенных алгоритмов, проверена на реальных данных из хранилища KEEL. Проведен сравнительный анализ с известными алгоритмами-аналогами «D-MOFARC» и «FARC-HD». Число правил, используемых классификаторами, построенными с помощью разработанных алгоритмов, значительно меньше числа правил в классификаторах-аналогах при сопоставимой точности классификации, что указывает на возможно более высокую интерпретируемость классификаторов, построенных с использованием предлагаемого подхода.

**Ключевые слова:** нечеткий классификатор, оптимизация параметров, алгоритм обезьян, формирование базы правил.

**Цитирование:** Hodashinsky I.A., Samsonov S.S. Design of fuzzy rule based classifier using the monkey algorithm // Business Informatics. 2017. No. 1 (39). P. 61–67. DOI: 10.17323/1998-0663.2017.1.61.67.

**Литература**

- Garcia-Galan S., Prado R.P., Exposito M.J.E. Rules discovery in fuzzy classifier systems with PSO for scheduling in grid computational infrastructures // *Applied Soft Computing*. 2015. No. 29. P. 424–435.
- Gorzalczany M.B., Rudzinski F. A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability // *Applied Soft Computing*. 2016. No. 40. P. 206–220.
- Laha A. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring // *Advanced Engineering Informatics*. 2007. No. 21. P. 281–291.
- Zhao R., Chai C., Zhou X. Using evolving fuzzy classifiers to classify consumers with different model architectures // *Physics Procedia*. 2012. No. 25. P. 1627–1636.
- Setnes M., Kaymak U. Fuzzy modeling of client preference from large data sets: An application to target selection in direct marketing // *IEEE Transactions on Fuzzy Systems*. 2001. No. 9. P. 153–163.
- Meier A., Werro N. A fuzzy classification model for online customers // *Informatica*. 2007. No. 31. P. 175–182.
- Горбунов И.В., Ходашинский И.А. Методы построения трехкритериальных парето-оптимальных нечетких классификаторов // *Искусственный интеллект и принятие решений*. 2015. № 2. С. 75–87.
- Scherer R. Multiple fuzzy classification systems // *Studies in Fuzziness and Soft Computing*, Vol. 288. Berlin: Springer-Verlag, 2012.
- Ходашинский И.А. Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных // *Информационные технологии*. 2012. № 3. С. 14–20.
- Antonelli M., Ducange P., Marcelloni F. An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets // *Neurocomputing*. 2014. No. 146. P. 125–136.
- Fazzolari F., Alcalá R., Herrera F. A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-MOFARC algorithm // *Applied Soft Computing*. 2014. No. 24. P. 470–481.
- Ходашинский И.А., Горбунов И.В. Оптимизация параметров нечетких систем на основе модифицированного алгоритма пчелиной колонии // *Мехатроника, автоматизация, управление*. 2012. № 10. С. 15–20.
- Ходашинский И.А., Дудин П.А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии // *Искусственный интеллект и принятие решений*. 2011. № 3. С. 26–33.
- Ходашинский И.А., Дудин П.А. Параметрическая идентификация нечетких моделей на основе гибридного алгоритма муравьиной колонии // *Автоматизация*. 2008. № 5 (44). С. 24–35.
- Zhao R., Tang W. Monkey algorithm for global numerical optimization // *Journal of Uncertain Systems*. 2008. No. 2. P. 165–176.
- Zheng L. An improved monkey algorithm with dynamic adaptation // *Applied Mathematics and Computation*. 2013. No. 222. P. 645–657.